Defining Short and Usable Processes©

Timothy G. Olson
Quality Improvement Consultants, Inc.

Many processes and procedures are large or difficult to use. The situation becomes even worse when complexity is involved. Putting large or difficult-to-use documentation on a Web site does not usually solve the problem. This article describes best practices for defining short and usable processes and procedures. These best practices have been used at real organizations over the last few years to define short and usable processes and procedures. Measurable results include cutting organizational processes and procedures in half while making them more usable (e.g., reducing 600 pages to 300 pages). The objectives of this article are to describe common problems with process documentation, including some human aspects of using process documents, discuss some best practices for defining short and usable processes and procedures, describe some success stories in real organizations, and provide some lessons learned.

Before we get into describing industry best practices, it is good to understand common problems with process documentation. Table 1 lists a summary of common problems.

How do we address the common problem with process documentation? One place to start is to recognize that not all documentation is used the same way. In this article, the term *process documentation* is used interchangeably with *policies, standards, processes, and procedures*. Table 2 contains a list of the types of process documentation and the ways in which these are used.

Table 1: *Common Problems Process Documentation*

Problem	Description
1. Too Big	Most process documentation is too big. Blaise Pascal once said, "I have made this letter longer than usual because I lack the time to make it shorter." This quote applies to most processes and procedures. Process documentation should be short, concise, and usable.
2. Not Enough Pictures	Most processes and procedures lack pictures and diagrams. If a picture is worth a thousand words, process documentation should have more pictures. Good process documentation should be a mixture of pictures and words. The best pictures are well-thought-out diagrams. The best diagrams for process documentation are process models.
3. Poorly Designed Documentation	Processes and procedures usually violate documentation design and good writing principles. Principles such as chunking[1] (i.e., seven plus or minus two), consistency, etc., are usually not used [1].
4. Unusable and *One Size Fits All*	Most processes and procedures are not designed with customers and users in mind, making them hard to use. Much documentation also has the *one-size-fits-all* mentality because it does not consider expert, intermediate, and beginner users.
5. Mixed Information Types	Policies, standards, processes, procedures, and training are all different types of information [2]. Most process documentation mixes these different types of information into the same paragraphs as if they were all used the same way. Each one of these document types has a different usage scenario.
6. Written Sequentially	Process documentation is not a novel, and is not meant to be read linearly (i.e. from beginning to end). Process documentation is reference material that is meant to be used non-linearly. This is why labeling is critical so that users can find information quickly [1].
7. Hard-to-Find Information Fast	Users of documentation will look for information for a few minutes. But if they cannot find the information quickly, many times they will give up in frustration and not use the processes or procedures. This can lead to serious nonconformance problems for many organizations.
8. Shelfware	Most process documentation becomes shelfware (i.e. collects dust on a shelf). Online processes (e.g. on an intranet) must be well designed or they will also become unused Web-ware.

Figure 1 [2] identifies the types of process documents and some critical relationships among those documents.

Process Documentation Usage Modes

Processes and procedures have different levels of users [3]. Some users have never used the process (i.e., beginner users). Some users have used the process a few times, but need guidance and lessons learned (i.e., intermediate users). Some users have used the process many times and may even be responsible for running the process (i.e., experts). The next sections will describe the three levels of documentation: expert, intermediate, and beginner.

Expert Mode Documentation

Expert mode documentation is short and concise [3]. When a pilot flies an airplane, he or she does not pull out a training manual; they use expert checklists for takeoff and landing. Expert mode documentation is made for experts and does not contain any training material. See Figure 2 (page 26) for an example of expert mode.

Most people want expert mode documentation because it is short. The problem with expert mode documentation is that not everybody is an expert. For example, not everyone can read a checklist for a rocket scientist (sometimes you really need to be a rocket scientist). Putting expert mode documentation in the hands of non-experts can be dangerous.

Why do experts need documentation if they are experts? Because people can forget things. This is why checklists are so powerful. Experts can also leave your organization, taking precious organizational knowledge with them. This is why expert knowledge should be documented.

Document Type	Usage
Policy	• Used by senior management to set direction in an organization. • States principles that organizations should follow.
Standard	• Specifies the sections of a document, and provides a description of what goes into those sections. • Makes the content of documents repeatable.
Process	• What happens over time to produce a desired result(s). • Should answer the five Ws: who, what, where, when, and why.
Procedure	• How-to or step-by-step information [1]. Example procedures are checklists, forms, and step/action tables. • Implements part of a process.

Table 2: *Types of Process Documentation and Their Uses*

Intermediate Mode Documentation

Intermediate mode documentation uses the expert mode documentation, but builds and adds to it by providing guidance and lessons learned. For example, guidance is very useful to people who do not have to follow a process or procedure very often. Even experts forget guidance and lessons learned for an annual process or an infrequently used process. Having guidance available to those who need or want it is very useful. Both Figure 2 and Table 3 (page 26) together are examples of intermediate mode.

Typically, guidance and lessons learned are not auditable. Process phases and procedure steps are required and auditable, but the supporting guidance and lessons learned are there for support only. One best practice is to distinguish between required steps and optional guidance. Some organizations have chosen to label guidance and lessons learned with a guidance label.

Beginner Mode Documentation

Beginner mode documentation uses the intermediate mode documentation, but adds training to it. Beginners should feel free to use the training manuals until they become familiar with the process. Beginners should also be mentored as appropriate. Some processes are simple, and some are complex. Complex processes should have formal training and be followed up by mentoring.

Usage Problem and Solutions

How can an organization afford to provide three versions of the same documentation? Someday software could allow this by just setting a documentation mode (expert/intermediate/beginner) and the user could see the appropriate information. A best practice that solves this problem is to define the process in *chunks* at the intermediate level (i.e., one version in intermediate mode). Add in training for the beginner and the expert can grab the appropriate chunks. Another best practice is that short, expert-mode documentation can also be provided for the experts.

What Is a Good Process?

The purpose of this section is to describe the required process elements necessary to define a *good process*. Table 4 (page 27) describes the practical *who, what*, *where*, *when*, *why*, and *how* process questions [2]. Each question is answered by a key process element. By addressing all the process elements on one page, the five W's (who, what, where, when, and why) can be represented in a diagram on one page in expert mode (Figure 2).

A good process should include the following:

- Address the five W's and answer the key process questions.
- Have both pictures and words (most people prefer pictures, but some people prefer words).
- Be usable and well written.
- Be well chunked and labeled so chunks can be found quickly [1].
- Be short (e.g., a diagram that answers the five W's on one page).

A Best Practice: Process Modeling

A good process should have pictures. It is said that a picture is worth a thousand words. However, not all pictures are good pictures. Some pictures cause confusion, and some pictures are more harmful than helpful. So what is a good picture? Process modeling is a best practice that helps design good diagrams that address the five W's. For a short and usable example, see Figure 2.

What is a process model? A process model *M* models *R* (reality) if *M* answers questions about *R* [4]. A good process model should answer the key process questions (i.e. the five W's in Table 4). A process model is typically represented by diagrams and powerful notations that represent roles, activities, work products, and the relationships between them.

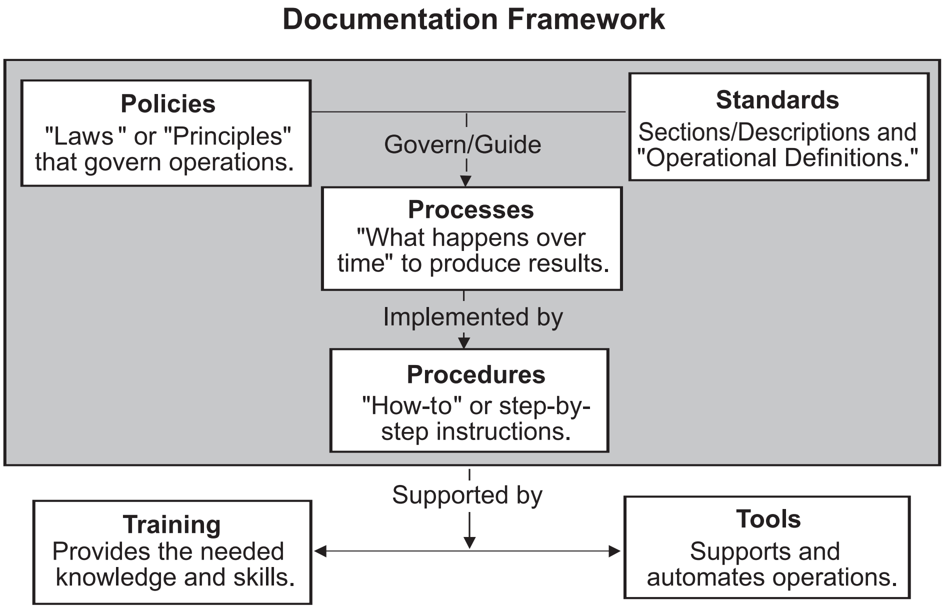
What Is a Good Procedure?

A good procedure consists of *how-to*, *step-by-step* information, and comes in three forms: checklists, forms, and step/action tables [3].

Checklists

Checklists are powerful repeatable repre-

Figure 1: *Types of Process Documents and Their Relationships*


Documentation Framework
Policies
"Laws" or "Principles" that govern operations.
Govern/Guide
Standards
Sections/Descriptions and "Operational Definitions."
Processes
"What happens over time" to produce results.
Implemented by
Procedures
"How-to" or step-by-step instructions.
Supported by
Training
Provides the needed knowledge and skills.
Tools
Supports and automates operations.

Note: Slide adapted from "A Software Process Famework for the SEI Capability Maturiy Model." Olson, et al. CMU/SEI-94-HB-01

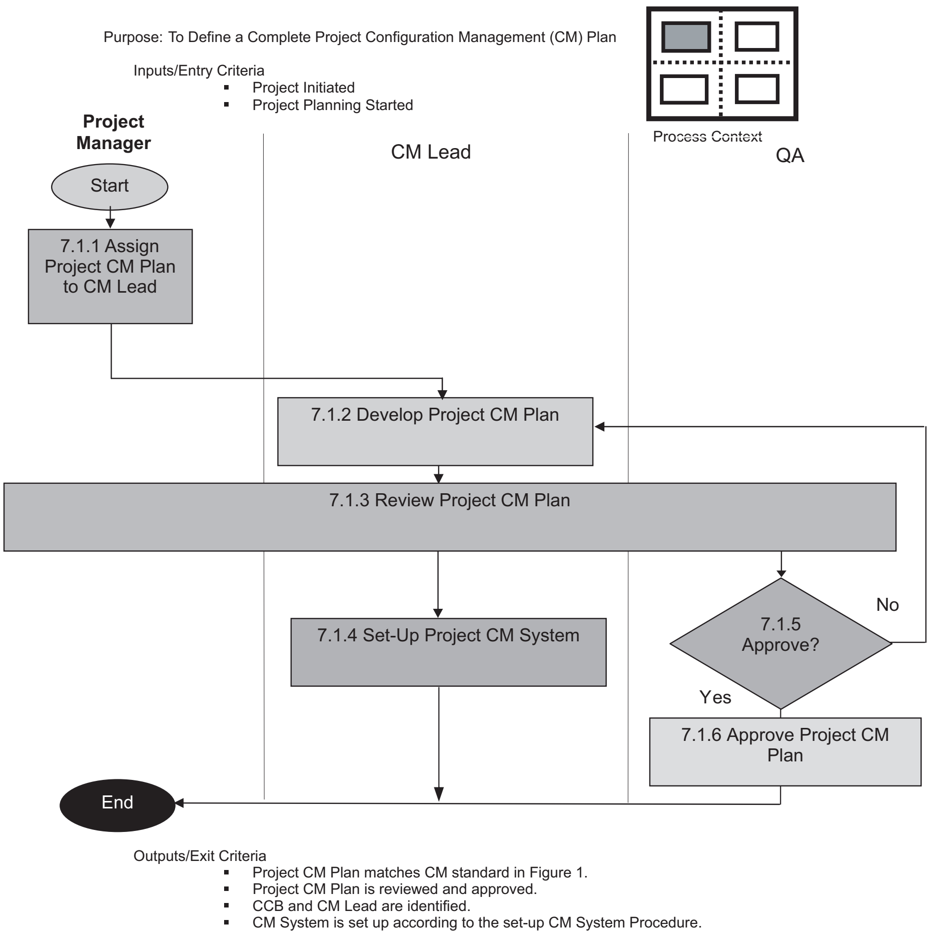

Purpose: To Define a Complete Project Configuration Management (CM) Plan
Inputs/Entry Criteria
Project Initiated
Project Planning Started
Process Context
Project Manager
CM Lead
QA
Start
7.1.1 Assign Project CM Plan to CM Lead
7.1.2 Develop Project CM Plan
7.1.3 Review Project CM Plan
7.1.4 Set-Up Project CM System
7.1.5 Approve?
No
Yes
7.1.6 Approve Project CM Plan
End
Outputs/Exit Criteria
Project CM Plan matches CM standard in Figure 1.
Project CM Plan is reviewed and approved.
CCB and CM Lead are identified.
CM System is set up according to the set-up CM System Procedure.

Figure 2: *7.1 Develop Project Configuration Management Plan (Expert Mode)*

Step	Role	Process Step
7.1.1	PM	**Assign Project CM Plan to CM Lead** At the appropriate time in the project (typically during project planning), the PM assigns the Project's CM plan development to the CM lead. **GUIDANCE:** A CM lead should have experience in setting up CM systems and performing CM. The CM Lead should also have been trained in CM.
7.1.2	CM Lead	**Develop Project CM Plan** The CM lead develops the Project CM plan according to the sections in the CM Plan Standard. It is required to follow the exact format of the CM Plan Standard **GUIDANCE:** The CM Plan Standard and example completed Project CM plans can be found on the Organizational Process Asset Web site.
7.1.3	All Active Roles	**Review Project CM Plan** The PM, CM Lead, and QA are required to peer review the CM Plan according to the CM Plan checklist in Appendix C. QA should review the CM plan against the CM Plan Standard. **GUIDANCE:** It is recommended to include the Project Team in the review.
7.1.4	CM Lead	**Set Up Project CM System** The CM Lead sets up the CM system for the project according to the Set-Up CM System Procedure in Appendix D. This step should be performed concurrently with steps 7.1.5 and 7.1.6. **GUIDANCE:** This task is only done once and may already be completed for maintenance projects.
7.1.5	QA	**Approve?** If "YES," then proceed to Process Step 7.1.6. If "NO," then procedd back to Process Step 7.1.2 for rework. After the third "No" iteration or 2 weeks after the first disapproval, escalate a CM Plan disapproval issue to Project Management. **GUIDANCE:** QA should ensure the CM Project Planning Process is followed. QA also needs to work with the Project Manager regarding approval issues.
7.1.6	QA	**Approve Project CM Plan** QA approves the Project CM Plan and places the plan on the project's official Web site. This sub-process is not complete until all steps have been completed (e.g., 7.1.4). **GUIDANCE:** QA should ensure the Project CM Plan is integrated into the overall Project Plan. The Project CM Plan is placed under CM with the Project Plan in the Project Planning Process.

Note: Project Manager (PM), Configuration Management (CM), Quality Assurance (QA).

Table 3: *7.1 Develop Project Configuration Management Plan (Intermediate Mode)*

sentations of activities that need to be completed to declare something completed. What makes checklists so powerful is that it usually does not matter in what order the checklist is completed. This is why checklists are useful for concurrent activities (e.g., versus flowcharts that are poor at representing concurrency).

Forms

Forms, along with instructions for completing the forms, are repeatable mechanisms for supporting processes. Forms are powerful mechanisms for collecting data in a repeatable way.

Step/Action Tables

One effective way to represent a procedure is using a step/action table [1]. Step/action tables are useful when order matters. For example, if a person needs to track his or her time, then starting to track time should not be the last step. For an example procedure of when order matters, see Table 5.

Some Success Stories

The best practices discussed in this article have been successfully used in practice over the last decade. More recently, major breakthroughs in defining extremely short and concise processes have also been achieved. An example of a short subprocess can be found in Figure 2 and Table 3. The following are some success story summaries (without revealing organizational identities).

Organizational Example 1

Organization No. 1 had a process that was not followed very well, and the process user feedback was that users did not like the process. The process had the following weaknesses (which are typical to most organizations):

- Mixture of document types: policies, standards, processes, and procedures.
- The existing processes lacked pictures (i.e., mostly text).
- The principle of *chunking* was violated in the flowcharts and in the number of process and procedure steps (e.g., processes and procedures with more than 20 steps), making the processes and procedures hard to use.
- Processes did not address all the five W's (e.g., *when* was missing; some other W's were also weak). It is hard to follow a process if you do not know when to start or when you are done.
- Procedures were very large and hard to follow (e.g., typical of International Organization of Standardization [ISO]

procedures).

New processes and procedures implemented the best practices described in this article and addressed all these weaknesses above. Process modeling was used to add good diagrams and address chunking of the flow chart. The new processes addressed all the five W's and were defined on one page for expert mode (Please see Figure 2 for an example of the five W's). In conclusion, organization No. 1 was much happier with diagrams on one page (i.e., process models), and with short, usable processes and procedures.

Key Process Question	Process Element
Why is the activity performed?	1. Purpose
What actions are performed?	2. Activities
What work products are used?	3. Input(s)
What work products are produced?	4. Output(s)
When does the activity begin?	5. Entry criteria
When does the activity end?	6. Exit criteria
Who performs activities?	7. Roles
Where is activity performed?	8. Process Context (e.g., Hierarchy)
How is the activity implemented?	9. Sub-Activity or Procedure

Table 4: *The Practical Process Questions*

Organizational Example 2

In organization No. 2, although detailed procedures existed, the overall processes had not been documented. Four sub-processes were defined on one-page diagrams (i.e., process models) for each process in expert mode (the five W's on one page), and a page of text along with guidance was developed in intermediate mode to support the diagram (see Figure 2 and Table 3 for a similar example). This organization packaged these four short processes in a single process guide in intermediate mode in about 20 pages total – four pages per process (similar processes documents can be more than 100 pages).

The experienced manager of this newly defined process was promoted in organization No. 2. Sometimes experienced people get stuck in positions because they are the only ones who know the processes, and often the processes are not documented. The new process document also helped with the transition because it allowed a new manager to come in and quickly learn the documented processes.

Organizational Example 3

Organization No. 3 requested a process review of all its processes that identified strengths, weaknesses, and made specific recommendations to senior management. During the organizational process review, it was discovered that a key process was not documented. The process was designed into three sub-processes, and each sub-process was defined on one page using a diagram (i.e., expert mode), and a page of text (along with guidance) was developed in intermediate mode to support the diagram (see Figure 2 and Table 3 for a similar sub-process example). The entire new process guide (including the policy, standard, process, and procedures) is about 20 pages long total (this includes about a dozen sections, including purpose, audience, usage, scope, metrics, procedures, references, etc). The new process also met all ISO requirements, which added about five pages. In conclusion, organization No. 3 is much happier and now has more complete, better, shorter, and more usable processes and procedures.

Organizational Example 4

Organization No. 4 complained that its processes and procedures were too large and difficult to use. After applying the best practices described in this article, the processes and procedures were cut in half (e.g., 600 total pages were reduced to 300 total pages). The processes and procedures are also more usable. What is fascinating about this example is although the processes and procedures were cut in half, no information was lost.

Some Lessons Learned

Here are some of the lessons learned while defining processes with best practices:

- Do not mix policy, standard, process, and procedure information (e.g., in the same paragraph). Label this different information, and consider how the information is used.
- Define all process documentation as simply as possible, but no simpler (information that is too simple does not work). Keep process documentation concise (i.e., short and sweet), but expect some processes to be complex.
- Use good pictures (most people prefer pictures). Process modeling is a best practice and scales up to very complex systems. Use process modeling to develop good pictures.
- For each process or sub-process, define the five W's on one page using a diagram (see Figure 2 for an example). A good process diagram can replace 20-25 pages of text.
- Use procedures (i.e., checklists, forms, and step/action tables [1]) for implementing processes and for repeatability.
- Use chunking (i.e., seven plus or minus two), organize the chunks, and label the chunks (so users can find information quickly). Process modeling and information mapping [1] help

Table 5: *Example: Step/Action Table Procedure*

Step	Action
1	Begin to track time (e.g., write down the start time).
2	Look for defects in the selected work product by using the appropriate data-driven checklist.
3	Log the defects of the Defect Form. Continue logging defects until the work product is completely inspected using the checklist.
4	End tracking time (e.g., write down the end time). Calculate the total time spent looking for and logging defects, and record the total time on the Defect Form.

tremendously with this principle.

- Account for beginner, intermediate, and expert users of the process.
- Design measurement into the process. Do not add measurement as an afterthought.
- The processes must be tailored to each organization, each business unit or division, and each project.

Summary

In summary, the objectives of this article are the following:

1. Describe common problems with process documentation, including some human aspects of using process documents.
2. Discuss some best practices for defining short and usable processes and procedures.
3. Describe some success stories in real organizations.
4. Provide some lessons learned.

Defining short and usable processes and procedures is challenging. There are many best practices that can be used to help improve process documentation. The approach summarized in this article uses a collection of best practices, all wrapped into a process (for defining processes). The author hopes that the readers have benefited from the description of some of the best practices along with the example process in Figure 2 and Table 3.◆

Note

[1] *Chunking* example: Most people can only remember *chunks* of information (e.g., that is why 15-16 digit credit cards numbers are broken into smaller chunks [seven, plus or minus two]; a 16-digit Visa number is usually broken into four groups of four digits).

References

1. Horn, Robert E. Mapping Hypertext: Analysis, Linkage, and Display Knowledge for the Next generation of On-Line Text and Graphics. Lexington, MA: The Lexington Institute, 1989.
2. Olson, Timothy G., et al. "A Software Process Framework for the SEI Capability Maturity Model." CMU/SEI-94-HB-01. Pittsburgh, PA: Software Engineering Institute, 1994.
3. Olson, Timothy G. "Defining Software Processes in Expert Mode." Software Engineering Process Group Conference, Atlanta, GA, 1998.
4. Ross, Douglas T., and Kenneth E. Schoman Jr. "Structured Analysis for Requirements Definition." IEEE Transactions on Software Engineering SE-3 (1) (1997): 6-15.

About the Author

Timothy G. Olson is founder and president of Quality Improvement Consultants, Inc. While performing quality consulting, Olson has helped organizations measurably improve quality and productivity, save millions of dollars in costs of poor quality, and reach higher Software Engineering Institute maturity levels. He has been formally trained in Crosby, Deming, Juran, International Organization of Standardization, Capability Maturity Model® (CMM®), CMM Integration^{SM}, and Six Sigma quality approaches. He is currently a senior member of the American Society of Quality and a member of the Institute of Electrical and Electronics Engineers. He has a master's degree from the University of Massachusetts.

Quality Improvement
Consultants, Inc.
7117 Obelisco CIR
San Diego, CA 92009
Phone: (760) 804-1406
Fax: (760) 804-1406
E-mail: tim.olson@qic-inc.com

Web Sites

Earned Schedule

www.earnedschedule.com

Earned Schedule (ES) introduces the concept of adding ES into a project management repertoire. ES discusses the difficulty of measuring schedule performance in dollars instead of units of time. ES is derived from Earned Value Management (EVM), but explores alternatives to EVM schedule tracking. It is derived from and is an extension to EVM. No additional data is needed for acquiring the ES measures; only the data from EVM is needed. In contrast to the cost-based indicators from EVM, the ES schedule performance indicators are time-based, making them easier to comprehend. The ES indicators provide a status and predictive ability for schedule, analogous to the facility for cost using EVM.

Software Technology Support Center

www.stsc.hill.af.mil

The U.S. Air Force's Software Technology Support Center (STSC) provides the Guidelines for Successful Acquisition and Management of Software Intensive Systems at <www.stsc.hill.af.mil/resources/tech_docs/gsam4.html>. A streamlined version of the content is provided and broken up into chapters for desktop usable access, as well as an overview of important software acquisition and development topics, helpful checklists for rapid self-inspection, and acquisition and management of software pointers. The STSC also oversees CrossTalk, offering monthly articles aimed at the software technology industry.

Software Engineering Institute

www.sei.cmu.edu

Carnegie Mellon University's Software Engineering Institute (SEI) has had the national mandate to advance the state of the practice of software engineering and to serve as a national resource in software engineering and technology. SEI aims to help organizations and individuals improve their software engineering management practices. SEI provides services dealing with the major categories of software engineering, including management, engineering and acquisitions. The Web site also provides links to education, training, and frequently asked questions in its mission to support software engineering professionals and advance software engineering and related disciplines to ensure the development and operation of systems with predictable and improved cost, schedule, and quality.